



US009237135B2

(12) **United States Patent**
Thom et al.

(10) **Patent No.:** **US 9,237,135 B2**
(45) **Date of Patent:** **Jan. 12, 2016**

(54) **TRUSTED AND CONFIDENTIAL REMOTE
TPM INITIALIZATION**

(71) Applicant: **Microsoft Corporation**, Redmond, WA
(US)

(72) Inventors: **Stefan Thom**, Snohomish, WA (US);
Kevin M. Litwack, Seattle, WA (US);
Shon Eizenhoefer, Bothell, WA (US);
Erik L. Holt, Sammamish, WA (US);
Yash Gandhi, Nagpur (IN)

(73) Assignee: **MICROSOFT TECHNOLOGY
LICENSING, LLC**, Redmond, WA
(US)

(*) Notice: Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/091,145**

(22) Filed: **Nov. 26, 2013**

(65) **Prior Publication Data**

US 2014/0089664 A1 Mar. 27, 2014

Related U.S. Application Data

(63) Continuation of application No. 12/249,042, filed on
Oct. 10, 2008, now Pat. No. 8,607,065.

(51) **Int. Cl.**
H04L 29/06 (2006.01)
H04L 9/32 (2006.01)
G06F 21/57 (2013.01)

(52) **U.S. Cl.**
CPC **H04L 63/0428** (2013.01); **G06F 21/57**
(2013.01); **G06F 21/575** (2013.01); **H04L**
9/3234 (2013.01); **H04L 9/3236** (2013.01);
H04L 2209/127 (2013.01)

(58) **Field of Classification Search**

CPC H04L 63/0428; H04L 9/3236; H04L
2209/127; H04L 9/3234

USPC 713/187, 168
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,299,354 B2	11/2007	Khanna
7,484,099 B2	1/2009	Bade
7,640,593 B2	12/2009	Tarkkala
7,900,058 B2	3/2011	Mabayoje
8,312,271 B2	11/2012	Bangerter
2005/0144440 A1	6/2005	Catherman
2005/0163317 A1	7/2005	Angelo
2006/0095505 A1	5/2006	Zimmer
2006/0184785 A1	8/2006	Challener
2006/0242428 A1	10/2006	Tarkkala
2007/0226496 A1	9/2007	Maletsky
2008/0046752 A1*	2/2008	Berger et al. 713/186
2008/0060068 A1	3/2008	Mabayoje

(Continued)

OTHER PUBLICATIONS

Goldman et al. (Linking Remote Attestation to Secure Tunnel
Endpoints, Nov. 3, 2006, 4 pages).*

(Continued)

Primary Examiner — Saleh Najjar

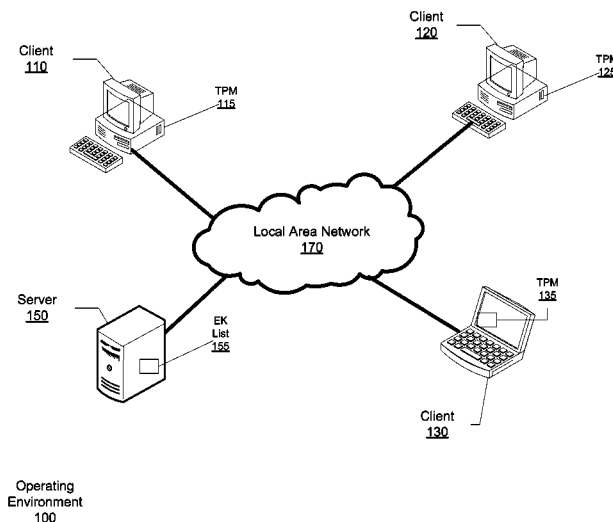
Assistant Examiner — Oleg Korsak

(74) *Attorney, Agent, or Firm* — Judy Yee; Timothy Churna;
Micky Minhas

(57) **ABSTRACT**

Techniques are provided to allow remote initialization of a
Trusted Platform Module. The results may be trusted and
confidential even if the target device has malicious operating
system or other software running.

18 Claims, 4 Drawing Sheets



(56)

References Cited

U.S. PATENT DOCUMENTS

2008/0184028 A1 7/2008 Anson
2009/0276620 A1 11/2009 McCarron

OTHER PUBLICATIONS

Wiese (Preliminary analysis of a trusted platform module (TPM) initialization process, Jun. 2007, 154 pages).*

Trusted Computing Group (Protection Profile PC Client Specific Trusted Platform Module TPM Family 1.2; Level 2 Version: 1.1; Jul. 10, 2008, 133 pages).*

CN Notice on the Second Office Action for Application No. 200980140559.1, Mar. 14, 2014.

"Windows Trusted Platform Module Management Step-by-Step Guide", Retrieved Apr. 22, 2008, pp. 8.

Sandra Rouane, "Initializing a TPM for First Use", Retrieved Mar. 24, 2007, pp. 3.

"HP Protect Tools Client Security Solutions Manageability for Customers with Limited IT Resources", Retrieved Mar. 2008.

Durfee, "Posture-Based Data Protection", Retrieved on or before Dec. 10, 2008, pp. 20.

"CCMV20-OV Client Configuration Mgr 2.0", Retrieved on or before Dec. 10, 2008, pp. 4.

Ulrich Kuhn, "Secure Data Management in Trusted Computing"; Cryptographic Hardware and Embedded Systems—CHES 2005 Lecture Notes in Computer Science, 2005, Retrieved Nov. 20, 2011.

David Safford, "Take control of TCPA" Linux Journal archive vol. 2003 Issue 112, Aug. 2003.

M. Strasser, A software-based TPM emulator for Linux Semesterarbeit, ETH Zurich, 2004, Retrieved Dec. 10, 2012.

CN Notice on the First Office Action for Application No. 200980140559.1, Jun. 27, 2013.

Search Report, for Application No. PCT/US2009/059846, mailed Apr. 28, 2010.

Schmidt, "On the deployment of Mobile Trusted Modules", In: IEEE Wireless Communications and Networking Conference, Mar. 31, 2007.

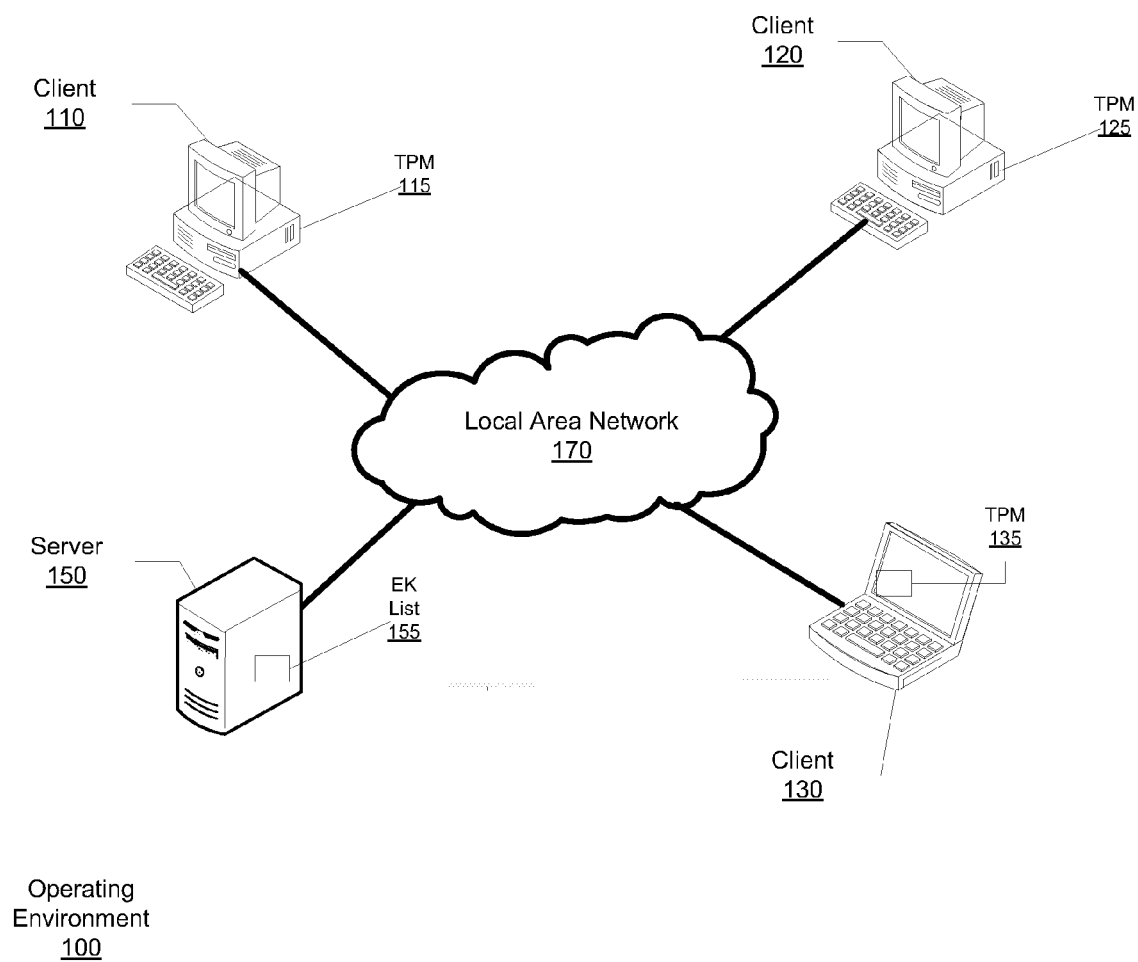
Kuhn, "Secure Data Management in Trusted Computing", In: Cryptographic Hardware and Embedded Systems, Aug. 29, 2005.

Congdon, "Implementing IEEE 802.1AR Secure Device Identity with the Trusted Computing Group's TPM", University of California, Davis, ECS 235A paper. Fall 2007.

EP Communication and Supplementary European Search Report for Application No. 09819820.3—1870 / 2335375 PCT/US2009059846, Reference FB23470, Apr. 28, 2015.

Kinney, "Trusted Platform Module Basics" Chapters 7-13, Dec. 31, 2006.

* cited by examiner

**FIG. 1**

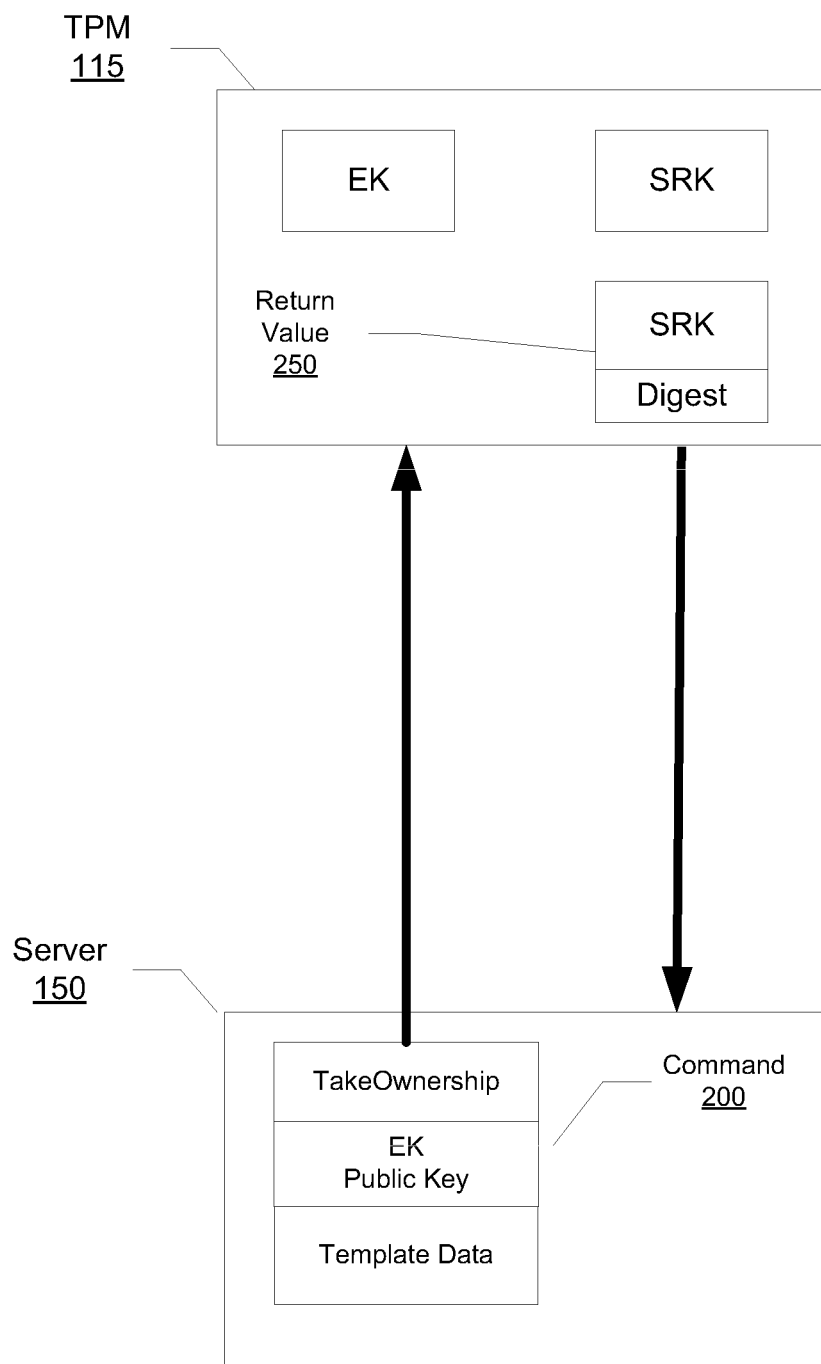
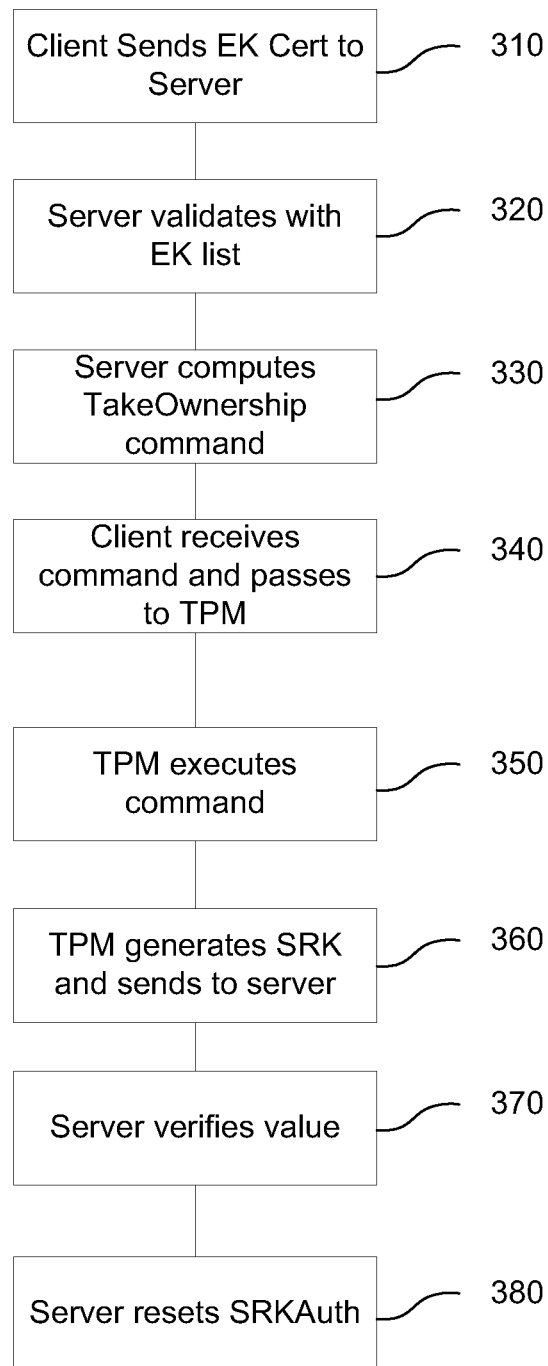
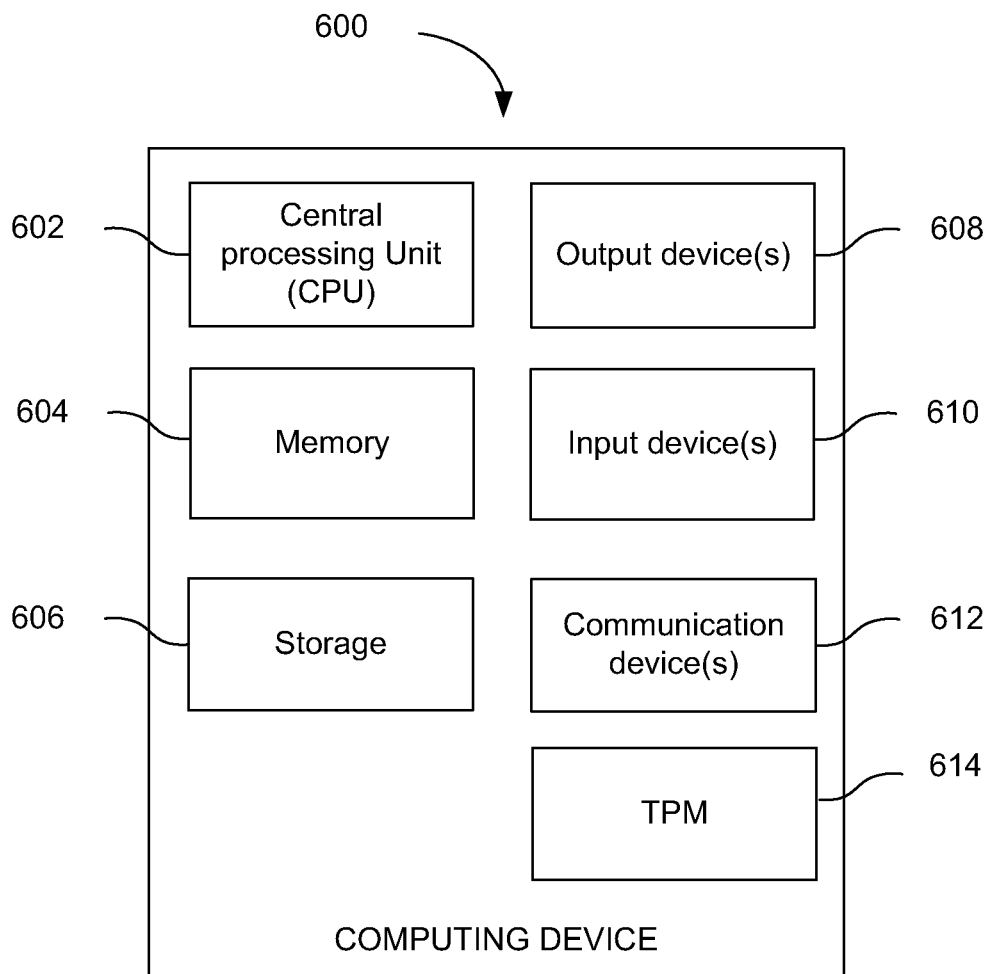


FIG. 2

**FIG. 3**

**FIG. 4**

1

TRUSTED AND CONFIDENTIAL REMOTE TPM INITIALIZATION

RELATED APPLICATION

This application is a Continuation of and claims benefit from U.S. patent application Ser. No. 12/249,042 that was filed on Oct. 10, 2008, and that is incorporated herein by reference in its entirety.

BACKGROUND

Trusted Platform Modules (TPMs) are hardware micro-controllers that store cryptographic keys, passwords, and digital certificates. TPMs also provide capabilities that allow authentication of devices, such as desktop computers, laptops, or cellular telephones. Once initialized, TPMs may be used to generate digital signing and encryption keys, allowing confirmation that messages are from the device that the message claims to be from. TPMs support commands including, for example, Init to initialize a TPM, TakeOwnership to set the owner value, and CreateEndorsementKeyPair to internally generate a key pair.

TPMs are often used in enterprise environments to allow systems to verify the source of information. TPMs must be securely and confidentially initialized before the keys and services provided by the device can be trusted.

SUMMARY

Described herein are, among other things, techniques for securely and confidentially initializing trusted platform modules remotely. For example, if a TPM is not securely and confidentially initialized, a malicious machine or operating system could host a emulated TPM in software and thereby deceive users and applications into thinking they can rely upon real TPM trust properties. A TPM emulated in this way may, for example, e, allow malicious software to gain access to secret cryptographic key material, or falsely report system configuration.

In one implementation, secure communications may be enabled by using a key provided by a device manufacturer. The secure communications may then allow a system administrator to perform a TakeOwnership operation remotely, rather than executing it locally on each device and risking exposure of sensitive information to the local software.

DESCRIPTION OF THE DRAWINGS

The detailed description provided below in connection with the appended drawings is intended as a description of example implementations and is not intended to represent the only forms in which a trusted and confidential remote initialization of a TPM may be performed. The description sets forth the functions of example implementations and the sequence of steps for constructing and operating the examples. However, the same or equivalent functions and sequences may be accomplished by alternate implementations.

The present description will be better understood from the following detailed description read in light of the accompanying drawings, wherein:

FIG. 1 is an example of an operating environment in which trusted and confidential remote TPM initialization may be implemented.

FIG. 2 provides additional detail for parts of FIG. 1, and adds an indication of an example data flow.

2

FIG. 3 is a flow chart showing an example of an implementation of trusted and confidential remote TPM initialization.

FIG. 4 illustrates a component diagram of a computing device according to one embodiment.

DETAILED DESCRIPTION

Described herein are, among other things, examples of various technologies and techniques that allow trusted and confidential remote TPM initialization. Although the examples are described and illustrated herein as being implemented in a personal computer system, the system described is provided as an example and not a limitation. As those skilled in the art will appreciate, the present examples are suitable for application in a variety of different types of systems.

In the figures, like reference numerals are used throughout several drawings to refer to similar components.

FIG. 1 is an example of an operating environment 100 in which trusted and confidential remote TPM initialization may be implemented. Server 150 contains an Endorsement Key list 155 provided by the manufacturers of Clients 110, 120, 130. The EK list 155 allows Server 150 to safely communicate over Local Area network 170 to remotely initialize TPMs 115, 125, 135.

Further details may be seen in FIG. 2, which shows an example of data flow between Server 150 and TPM 115. In this example, Server 150 uses a public key for TPM 115's EK to encrypt template data as part of TakeOwnership command 200, and submits it to TPM 115.

While this example uses an Endorsement Key list 155 provided by the manufacturers of Clients 110, 120, 130, in other implementations it could be an individual certificate from a trusted TPM vendor. One skilled in the art will realize that there may be several techniques used to authenticate the TPM.

FIG. 3 is a flow chart providing more detail on one possible implementation of this process. A client initiates the process by sending 310 an EK certificate from a TPM to a server. The server validates 320 the certificate using EK information provided by the client's manufacturer. The server then computes 330 a TakeOwnership command, including server-specified values for OwnerAuth and SRKAuth for the TPM, and encrypts the command with the public key portion of the TPM's EK. The server then sends this encrypted command to the client, which passes it 340 to the TPM. The client software cannot read the data because it is encrypted with a protected TPM key.

The TPM then decrypts the encrypted data using the EK private key, and executes 350 the TakeOwnership command. The TPM then creates a Storage Root Key (SRK), calculates a keyed-Hash Message Authentication Code (HMAC), and sends 360 that data back to the server. Because it is just a digest, the client cannot read it even if the client is malicious. The server can verify the HMAC, since it contains the shared secret of the SRK, and trust that the data is confidential and is from the TPM specified by the EK certificate obtained from the manufacturer by using 370 the EK provided by the manufacturer.

At this point in the process, the client cannot use or create keys on the TPM because the SRK is set to a secret value. To resolve this, the server initiates an encrypted tunnel to the TPM using the SRK public key. The client cannot pretend to be the TPM since it does not have the SRK private key. Once the tunnel is established, the server can generate keys, create identities, and set up the delegation tables. The server may also generate a trusted signing key that can be used for the

3

TPM operation CertifyKey, which will allow the server to confirm that any future key is, in fact, from the TPM.

Once the server has completed the operations desired for initializing the TPM, it will reset **380** SRKAuth back to the well-known value of all zeroes, so that the client can use the TPM to generate and use keys, but preventing the client from performing Owner privileged operations. After the initial configuration is successful, the server may also re-open the encrypted channel and modify the configuration of the TPM at a later time.

FIG. 4 illustrates a component diagram of a computing device according to one embodiment. The computing device **600** can be utilized to implement one or more computing devices, computer processes, or software modules described herein. In one example, the computing device **600** can be utilized to process calculations, execute instructions, receive and transmit digital signals. In another example, the computing device **600** can be utilized to process calculations, execute instructions, receive and transmit digital signals, receive and transmit search queries, and hypertext, compile computer code as required by Server **150** or Clients **110**, **120**, **130**.

The computing device **600** can be any general or special purpose computer now known or to become known capable of performing the steps and/or performing the functions described herein, either in software, hardware, firmware, or a combination thereof.

In its most basic configuration, computing device **600** typically includes at least one central processing unit (CPU) **602** and memory **604**. Depending on the exact configuration and type of computing device, memory **604** may be volatile (such as RA), non-volatile (such as ROM, flash memory, etc.) or some combination of the two. Additionally, computing device **600** may also have additional features/functionality. For example, computing device **600** may include multiple CPU's. The described methods may be executed in any manner by any processing unit in computing device **600**. For example, the described process may be executed by both multiple CPU's in parallel.

Computing device **600** may also include additional storage (removable and/or non-removable) including, but not limited to, magnetic or optical disks or tape. Such additional storage is illustrated in FIG. 6 by storage **206**. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Memory **604** and storage **606** are all examples of computer storage media. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computing device **600**. Any such computer storage media may be part of computing device **600**.

Computing device **600** may also contain communications device(s) **612** that allow the device to communicate with other devices. Communications device(s) **612** is an example of communication media. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limi-

4

tation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RE, infrared and other wireless media. The term computer-readable media as used herein includes both computer storage media and communication media. The described methods may be encoded in any computer-readable media in any form, such as data, computer-executable instructions, and the like.

Computing device **600** may also have input device(s) **610** such as keyboard, mouse, pen, voice input device, touch input device, etc. Output device(s) **608** such as a display, speakers, printer, etc. may also be included. All these devices are well known in the art and need not be discussed at length.

Computing device **600** may also have a Trusted Platform Module (TPM).

Those skilled in the art will realize that storage devices utilized to store program instructions can be distributed across a network. For example, a remote computer may store an example of the process described as software. A local or terminal computer may access the remote computer and download a part or all of the software to run the program. Alternatively, the local computer may download pieces of the software as needed, or execute some software instructions at the local terminal and some at the remote computer (or computer network). Those skilled in the art will also realize that by utilizing conventional techniques known to those skilled in the art that all, or a portion of the software instructions may be carried out by a dedicated circuit, such as a DSP, programmable logic array, or the like.

The invention claimed is:

1. A method performed on a computing device comprising a Trusted Platform Module ("TPM"), the method comprising:
 - receiving, from a remote device, a command for taking ownership of the TPM;
 - creating, by the TPM in response to the received command, a key;
 - sending the created key to the remote device; and
 - establishing a tunnel between the remote device and the TPM, where the established tunnel is encrypted based on the sent key, where the TPM is configured for being remotely initialized by the remote device via the established tunnel, and where the TPM is further configured for allowing the remote device to modify a configuration of the TPM via the established tunnel at a later time.
2. The method of claim 1 where the TPM is further configured for allowing the remote device to use the TPM to generate keys via the established tunnel.
3. The method of claim 1 where the TPM is further configured for allowing the remote device to use the TPM to create identities via the established tunnel.
4. The method of claim 1 where the TPM is further configured for allowing the remote device to use the TPM of the computing device to set up delegation tables via the established tunnel.
5. The method of claim 1 where the TPM is further configured for allowing the remote device to re-open the established tunnel at a later time.
6. The method of claim 1 where the TPM is further configured for allowing the computing device to use the TPM, but not to use the TPM to perform privileged operations.
7. A system comprising:
 - a computing device;
 - memory that comprises at least one program module; and
 - a Trusted Platform Module ("TPM"), the system configured according to the at least one program module to:
 - receive, from a remote device, a command for taking ownership of the TPM;

5

create, by the TPM in response to the received command, a key;

send the created key to the remote device; and

establish a tunnel between the remote device and the TPM, where the established tunnel is encrypted based on the sent key, and where the TPM is configured for being remotely initialized by the remote device via the established tunnel, and where the TPM is further configured for allowing the remote device to modify a configuration of the TPM via the established tunnel at a later time.

8. The system of claim 7 where the TPM is further configured for allowing the remote device to use the TPM to generate keys via the established tunnel.

9. The system of claim 7 where the TPM is further configured for allowing the remote device to use the TPM to create identities via the established tunnel.

10. The system of claim 7 where the TPM is further configured for allowing the remote device to use the TPM to set up delegation tables via the established tunnel.

11. The system of claim 7 where the TPM is further configured for allowing the remote device to re-open the established tunnel at a later time.

12. The system of claim 7 where the TPM is further configured for allowing the computing device to use the TPM, but not to use the TPM to perform privileged operations.

13. At least one computer storage device that comprises computer-executable instructions that, based on execution by a computing device that includes a Trusted Platform Module ("TPM"), configure cause the computing device to perform actions comprising:

6

receiving, from a remote device, a command for taking ownership of the TPM;

creating, by the TPM in response to the received command, a key;

sending the created key to the remote device; and

establishing a tunnel between the remote device and the TPM, where the established tunnel is encrypted based on the sent key, and where the TPM is configured for being remotely initialized by the remote device via the established tunnel, and where the TPM is further configured for allowing the remote device to modify a configuration of the TPM via the established tunnel at a later time.

14. The at least one computer storage device of claim 13 where the TPM is further configured for allowing the remote device to use the TPM to generate keys via the established tunnel.

15. The at least one computer storage device of claim 13 where the TPM is further configured for allowing the remote device to use the TPM to create identities via the established tunnel.

16. The at least one computer storage device of claim 13 where the TPM is further configured for allowing the remote device to use the TPM to set up delegation tables via the established tunnel.

17. The at least one computer storage device of claim 13 where the TPM is further configured for allowing the remote device to re-open the established tunnel at a later time.

18. The at least one computer storage device of claim 13 where the TPM is further configured for allowing the computing device to use the TPM, but not to use the TPM to perform privileged operations.

* * * * *